

Appendix

VBAの実用サンプル

□01 データの入力フォーム

Sample_Data/ap-01-01.xlsm

● データ入力用フォームの概要

ここでは、ワークシート上のレコードをユーザーフォームに読み込んで、ユーザーフォーム上で表示/編集/追加するサンプルを紹介します。

対象となるワークシートのセル **D1** には、「**=COUNTA(A4:A1048576)**」のように **COUNTA 関数**を設定してデータ件数を求めています。また、データ件数を表示するセル **D1** に名前「件数」を設定しています。

<Memo>

COUNTA 関数は引数に指定したセル範囲のデータの個数を求めます。

「=COUNTA (A4:A1048576)」は、セル A4 から一番下までのデータの個数を求めます。

■ 作成するユーザーフォーム

名前[件数]

① [顧客マスター] シートの
[フォーム表示] ボタンをクリックする

② [顧客管理] フォームを表示する。
画面上でデータの表示/修正/追加/検索
などの操作ができる

①でボタンに割り当てられている[フォーム表示]プロシージャ（作成場所：標準モジュール）が実行され、②のフォームが開きます。

【Sample】[フォーム表示]ボタンがクリックされた際の処理

Sub フォーム表示 ()

UserForm1.Show vbModeless 'ユーザーフォームを開く

End Sub

●ユーザーフォームの作成

次のようなユーザーフォームを作成します。（作成手順は p. 372 を参照してください）なお、文字を表示するだけのラベルの設定は省略しています。

■プロパティウィンドウでの設定内容

No.	ユーザーフォーム/コントロール名	プロパティ	設定値
1	ユーザーフォーム	(オブジェクト名)	UserForm1
		Caption	顧客管理
2	テキストボックス	(オブジェクト名)	txtNo
3	テキストボックス	(オブジェクト名)	txtName
4	コマンドボタン	(オブジェクト名)	btnKana
		Caption	フリガナ表示
5	テキストボックス	(オブジェクト名)	txtKana
6	フレーム	Caption	性別

No.	ユーザーフォーム/コントロール名	プロパティ	設定値
7	オプションボタン	(オブジェクト名)	optMan
		Caption	男
8	オプションボタン	(オブジェクト名)	optWoman
		Caption	女
9	リストボックス	(オブジェクト名)	lstKubun
10	コンボボックス	(オブジェクト名)	cmbYear
11	コンボボックス	(オブジェクト名)	cmbMonth
12	コンボボックス	(オブジェクト名)	cmbDay
13	テキストボックス	(オブジェクト名)	txtKensaku
14	コマンドボタン	(オブジェクト名)	btnKensaku
		Caption	顧客 NOで検索
15	コマンドボタン	(オブジェクト名)	btnTop
		Caption	先頭
16	コマンドボタン	(オブジェクト名)	btnPre
		Caption	前
17	コマンドボタン	(オブジェクト名)	btnNext
		Caption	次
18	コマンドボタン	(オブジェクト名)	btnLast
		Caption	最後
19	コマンドボタン	(オブジェクト名)	btnNew
		Caption	新規
20	ラベル	(オブジェクト名)	lblRec
21	ラベル	(オブジェクト名)	lblAll
22	コマンドボタン	(オブジェクト名)	btnSave
		Caption	登録
23	コマンドボタン	(オブジェクト名)	btnQuit
		Caption	終了 (Q)
		Accelerator	Q

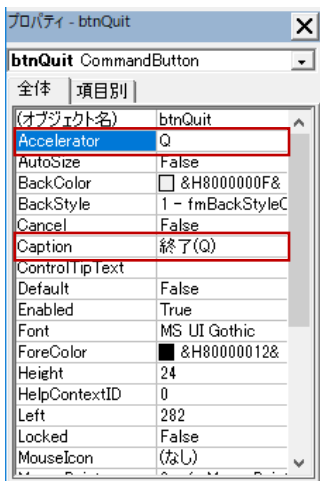
※ここでは男女のオプションボタンをフレーム内に配置しています。先にフレームを追加した後、その中にオプションボタン（⑦、⑧）を追加します。

＜コラム＞ コマンドボタンにアクセスキーを設定する

アクセスキーとは、コマンドボタンなどのコントロールを選択するための1文字です。

Altキーを押しながらアクセスキーを押すと、そのコントロールが選択されたり、コマンドを実行したりできます。コマンドボタンであれば、クリックと同じ操作になります。

アクセスキーを設定するには、[プロパティウィンドウ]の[**Accelerator**]プロパティでアクセスキーに割り当てたいアルファベットを半角文字で入力し、[**Caption**]プロパティで[終了 (Q)]のように割り当てたアクセスキーを入力します。



●開く/閉じるときのプロシージャを作成

フォームが開く直前に実行するユーザーフォームの **Initialize イベントプロシージャ**で初期設定をします。

ここでは、リストボックスとコンボボックスの**選択肢となる値を追加**します。

[終了] ボタン (btnQuit) をクリックしたときに実行する **Click イベントプロシージャ**でフォームを閉じる処理を行います。

【Sample】 Initialize イベントプロシージャ

```
Private Sub UserForm_Initialize()  
    Dim i As Long  
    'リストボックスに選択肢を追加  
    lstKubun.AddItem "プレミアム"  
    lstKubun.AddItem "ゴールド"  
    lstKubun.AddItem "レギュラー"  
    'コンボボックスに年の選択肢を追加  
    For i = 1930 To 2022  
        cmbYear.AddItem i  
    Next  
    'コンボボックスに月の選択肢を追加  
    For i = 1 To 12  
        cmbMonth.AddItem i  
    Next  
    'コンボボックスに日の選択肢を追加  
    For i = 1 To 31  
        cmbDay.AddItem i  
    Next  
End Sub
```

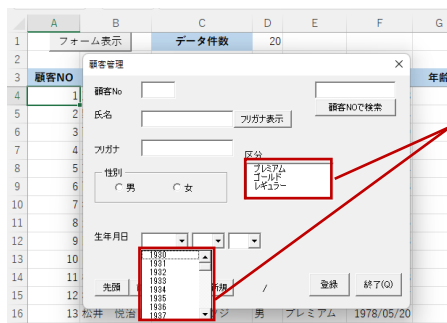
※データが1件もない場合の処理を p.16 で追加しています。

【Sample】 [終了] ボタン (btnQuit) がクリックされた際の処理

```
Private Sub btnQuit_Click()  
    Unload Me    'ユーザーフォームを閉じる  
End Sub
```

なお、この章で紹介するプロシージャ内には、エラー処理コードは記述していません。本編 p.346 を参考に必要に応じてエラー処理コードを追加してください。

■ユーザーフォーム



リストボックス、コンボボックスに選択肢が追加されている

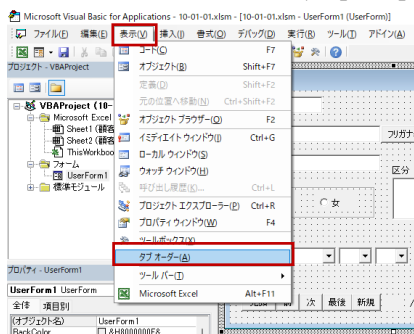
●タブオーダーの設定

ユーザーフォームを開いて、**Tab**キーを押すと、**コントロールを配置した順番にフォーカスが移動**します。この移動順のことを**タブオーダー**といいます。

フォーカスとは、**文字の入力などキーボード操作を受け取ることができる状態**で、テキストボックスではカーソルが表示され、オプションボタンやコマンドボタンでは表示されている文字列が点線で囲まれ、操作対象となっている状態です。タブオーダーをデータの入力順に設定しておけば入力がスムーズです。

タブオーダーを設定する手順は次の通りです。

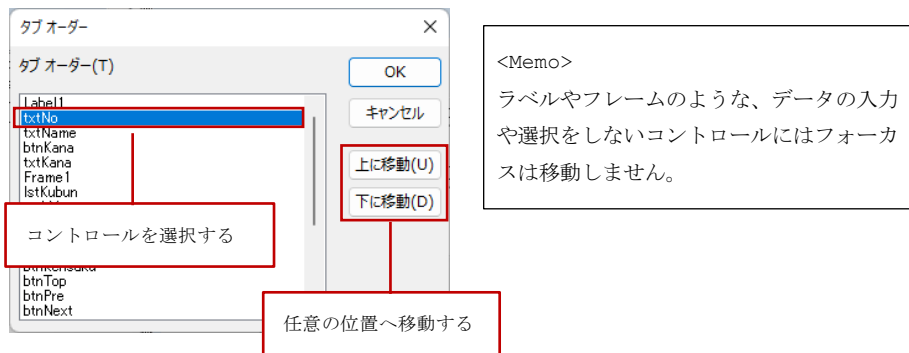
①ユーザーフォームを表示し、[表示] メニューの [タブオーダー] をクリック



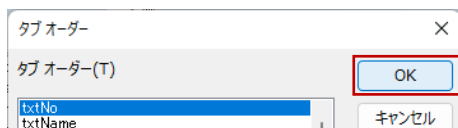
<Memo>

ユーザーフォームが表示されていない場合は、[プロジェクトエクスプローラー] で開きたいユーザーフォーム名をダブルクリックします。

② [タブオーダー] ダイアログで順番を移動したいコントロールを選択し、[上/下に移動] ボタンをクリックしてコントロールを任意の位置まで移動

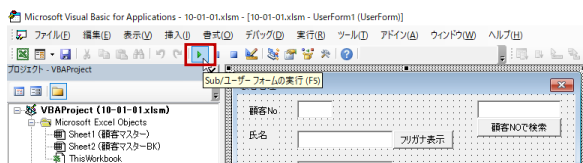


③ 目的の順番に変更できたら[OK]ボタンで確定



<Memo>
フレーム内に配置したオプションボタンの移動順は、フレームのタブオーダーで設定します。フレームを選択し、[タブオーダー] ダイアログボックスを表示して同様に順番を設定します。

④ [Sub/ユーザーフォームの実行] ボタンでユーザーフォームを開く



⑤ **Tab**キーを押して、タブオーダーで設定した順番にフォーカスが移動することを確認する。



<Memo>

逆方向に移動するには、**Shift**キーを押しながら**Tab**キーを押します。

●GetPhonetic メソッドでフリガナを取得する

漢字の氏名を入力したら、そのフリガナを自動で表示できるようにすると入力の手間が省けます。**VBA で入力された文字のフリガナを取得する**には Application オブジェクトの **GetPhonetic メソッド**を使います。

<書式>

Application.GetPhonetic ([Text])

引数 Text : フリガナに変換する文字列を指定

省略時は以前に指定した引数 [Text] のフリガナ候補文字列を返す候補となる文字列がない場合は空の文字列 [""] を返す

以下のコードは、**【フリガナ表示】ボタン**をクリックしたときに実行する **Click イベントプロシージャ**です。テキストボックス **txtName** に入力された文字列のフリガナをテキストボックス **txtKana** に表示します。

【Sample】 [フリガナ表示] ボタン (btnKana) がクリックされた際の処理

```
Private Sub btnKana_Click()  
    txtKana.Value = Application.GetPhonetic(txtName.Value)  
End Sub
```

- ①氏名を入力して[フリガナ表示]ボタンをクリックする。
- ②[フリガナ]欄にフリガナが表示される。

■ ユーザーフォームの実行結果

<Memo>

正しいフリガナが表示されなかった場合は、[フリガナ]欄で直接修正してください。

<コラム> フリガナをひらがなで表示する

GetPhonetic メソッドで表示されるフリガナをひらがなにしたい場合は、文字種を変換するStrConv関数を使用して以下のように記述します。

【Sample】

```
txtKana.Value = StrConv(Application.GetPhonetic(txtName.Value), vbHiragana)
```

<コラム> 日本語入力モードを設定する

テキストボックスに入力するデータの種別にあわせて日本語入力モードを設定できます。テキストボックスの [プロパティウィンドウ] の [IMEMode] プロパティで文字種を一覧から選択します。テキストボックスだけでなくコンボボックスでも設定できます。

■ IMEMode プロパティの主な設定値

設定値	内容
mIMEModeNoControl	変更なし (既定値)
fmIMEModeOn	IME オン
fmIMEModeOff	IME オフ (英語モード)
fmIMEModeDisable	IME オフ (使用不可)
fmIMEModeHiragana	全角ひらがな
fmIMEModeKatakana	全角カタカナ
fmIMEModeKatakanaHalf	半角カタカナ
fmIMEModeAlphaFull	全角英数字
fmIMEModeAlpha	半角英数字

●ユーザーフォームのコントロールにセルの値を表示する

テキストボックスなどユーザーフォーム上のコントロールにセルの値を表示するには、コントロールの **Value** プロパティか **Text** プロパティにセルの **Value** プロパティの値を代入します。オプションボタンの場合は、**True/False** の値とセルの値を対応させるための処理が必要になります。

■顧客マスター表のデータとユーザーフォームの対応

The screenshot shows a spreadsheet with columns A through G. Row 3 contains headers: 顧客NO, 氏名, フリガナ, 性別, 区分, 生年月日, 年齢. Row 4 contains data: 1, 正木 希海, マサキ ノゾミ, 女, レギュラー, 1995/01/18, 27. A user form is overlaid on the spreadsheet, with fields corresponding to the data in row 4. A red box highlights the '顧客NO' field in the form, which contains the value '1'. A red arrow points from this field to a text box on the right that says '顧客マスターの表のデータをユーザーフォームに表示する'.

顧客NO	氏名	フリガナ	性別	区分	生年月日	年齢
1	正木 希海	マサキ ノゾミ	女	レギュラー	1995/01/18	27
2	野本 聡	ノモト サトシ	男	ゴールド	1977/10/24	44

●セルの値を読み込むプロシージャを作成する

フォームを開いたときは1レコード目、[次] ボタンを押したら次のレコード… …というように、場合によって表示するレコードは異なります。**どのレコードを表示するかは、[顧客 NO] 列のように固有の値をもつ列の値を引数に使って、レコードを読み込む Sub プロシージャを作成し、それを呼び出して使用します。**

ここでは、[顧客 NO] の値を代入する引数 **cNo** を受け取って、レコードをフォームに読み込む Sub プロシージャ [レコード読み込み] を作成します。このプロシージャは **UserForm1** のコードウィンドウに記述します。

【Sample】 [UserForm1] ユーザーフォームのコードウィンドウに記述する処理

'A: [レコード読み込み] プロシージャ、変数宣言と値の代入

```
Dim rRng As Range ----- ①
Const S_ROW As Long = 3 '表の見出し行の行番号 ----- ②

Sub レコード読み込み(cNo As Long) 'cNo…顧客 NO の値 ----- ③
    Dim r As Long '顧客 No のセルの行番号代入用
    Set rRng = Range("A3").CurrentRegion.Columns(1).Find( _ ④
        What:=cNo, lookat:=xlWhole)
    If rRng Is Nothing Then
        MsgBox "データが見つかりません" ⑤
        Exit Sub
    End
    r = rRng.Row '見つかったセルの行番号を取得 ----- ⑥
```

'B: テキストボックスに値を表示

```
txtNo.Text = Cells(r, 1).Value '顧客 No
txtName.Text = Cells(r, 2).Value '氏名
txtKana.Text = Cells(r, 3).Value 'フリガナ ⑦
```

'C: オプションボタンに値を表示 (性別)

```
Select Case Cells(r, 4).Value
    Case "男"
        optMan.Value = True
    Case "女"
        optWoman.Value = True
    Case Else
        optMan.Value = False
        optWoman.Value = False
End Select ⑧
```

'D: リストボックスに値を表示 (区分)

```
If Cells(r, 5).Value <> "" Then
    lstKubun.Text = Cells(r, 5).Value
Else
    lstKubun.ListIndex = -1
End If
```

⑨

'E: コンボボックスに値を表示 (生年月日)

```
If Cells(r, 6).Value <> "" Then
    cmbYear.Text = Year(Cells(r, 6).Value)
    cmbMonth.Text = Month(Cells(r, 6).Value)
    cmbDay.Text = Day(Cells(r, 6).Value)
Else
    cmbYear.Text = ""
    cmbMonth.Text = ""
    cmbDay.Text = ""
End If
```

⑩

'F: ラベルに現在のレコード番号とレコード数を表示

```
lblRec.Caption = r - S_ROW
lblAll.Caption = "/" & Range("件数").Value
End Sub
```

⑪

ブロック A:

[レコード読み込み] プロシージャ、変数宣言と値の代入を行います。

'A: [レコード読み込み] プロシージャ、変数宣言と値の代入

```
Dim rRng As Range ----- ①
```

```
Const S_ROW As Long = 3 '表の見出し行の行番号 ----- ②
```

```
Sub レコード読み込み(cNo As Long) 'cNo…顧客NOの値 ----- ③
```

```
    Dim r As Long '顧客NOのセルの行番号代入用
```

```
    Set rRng = Range("A3").CurrentRegion.Columns(1).Find( _ ④
```

```
        What:=cNo, lookat:=xlWhole)
```

```

If rRng Is Nothing Then
    MsgBox "データが見つかりません" ⑤
Exit Sub

End

r = rRng.Row '見つかったセルの行番号を取得 ----- ⑥

```

- ①表示する顧客 No のセルを代入するために Range 型の変数 **rRng** をモジュールレベルで宣言する。
- ②表の見出し行の行番号 3 を代入した定数 **S_ROW** を宣言する。
- ③引数 **cNo**（顧客 No の値）を受け取って実行する「レコード読み込み」というプロシージャを作成する。
- ④セル **A3** を含む表の 1 列目（「顧客 No」の列）から引数 **cNo** の値を持つセルを検索し、見つかったら変数 **rRng** に代入する。
- ⑤見つからなかった場合は、「データが見つかりません」とメッセージ表示して終了する。
- ⑥変数 **rRng**（顧客 No のセル）の行番号を変数 **r** に代入する。

ブロック B:

テキストボックスに値を表示します。

```

'B: テキストボックスに値を表示
txtNo.Text = Cells(r, 1).Value '顧客No
txtName.Text = Cells(r, 2).Value '氏名
txtKana.Text = Cells(r, 3).Value 'フリガナ ⑦

```

- ⑦テキストボックスにそれぞれ、
- txtNo** → **r** 行 **1** 列目（顧客 No）の値を表示する。
 - txtName** → **r** 行 **2** 列目（氏名）の値を表示する。
 - txtKana** → **r** 行 **3** 列目（フリガナ）の値を表示する。

ブロック C:

オプションボタンの選択値を設定します。

'C: オプションボタンに値を表示 (性別)

```
Select Case Cells(r, 4).Value
    Case "男"
        optMan.Value = True
    Case "女"
        optWoman.Value = True
    Case Else
        optMan.Value = False
        optWoman.Value = False
End Select
```

⑧

⑧ **r** 行 **4** 列目のセルの値 (性別) が

男の場合 → オプションボタン **optMan** をオンにする。

女の場合 → オプションボタン **optWoman** をオンにする。

どちらでもない場合 → **optMan**、**optWoman** どちらもオフにする。

ブロック D:

リストボックスの選択値を設定します。

'D: リストボックスに値を表示 (区分)

```
If Cells(r, 5).Value <> "" Then
    lstKubun.Text = Cells(r, 5).Value
Else
    lstKubun.ListIndex = -1
End If
```

⑨

⑨ **r** 行 **5** 列目のセルの値 (区分) が、

空欄でない場合 → **r** 行 **5** 列目の値をリストボックス **lstKubun** に表示する。

空欄の場合 → **lstKubun** は未選択の状態にする。

<Memo>

セルの値をリストボックスの Text プロパティに代入することで、リストボックスの項目を選択できます。このときセルの値がリストボックスに追加されている必要があります。追加されていない場合はエラーになります。

ブロック E:

コンボボックスに値を表示します。

'E: コンボボックスに値を表示 (生年月日)

```
If Cells(r, 6).Value <> "" Then
    cmbYear.Text = Year(Cells(r, 6).Value)
    cmbMonth.Text = Month(Cells(r, 6).Value)
    cmbDay.Text = Day(Cells(r, 6).Value)
Else
    cmbYear.Text = ""
    cmbMonth.Text = ""
    cmbDay.Text = ""
End If
```

⑩

⑩ **r** 行 **6** 列目のセルの値 (生年月日) が、

空欄でない場合 → **r** 行 **6** 列目のセルの値から年/月/日を取得して各コンボボックスに表示する。

空欄の場合 → それぞれのコンボボックスを空欄にする。

<Memo>

コンボボックスを空欄にする際、Text プロパティに「""」を代入する代わりに「cmbYear.ListIndex=-1」と記述することもできます。

ブロック F:

フォームに表示するデータのレコード番号は、表のデータの行番号から見出し行の行番号 (**S_ROW**) を引いた数となります。これを使って、ラベルに現在のレコード番号とレコード数を表示します。

'F: ラベルに現在のレコード番号とレコード数を表示

```
lblRec.Caption = r - S_ROW
lblAll.Caption = "/" & Range("件数").Value
```

⑪

⑪ ラベルに現在のレコード番号、レコード数をそれぞれ表示

ラベル **lblRec** (レコード番号)

→ 変数 **rRng** の行番号 **r** から定数 **S_ROW** (**3**) を引いた値を表示する。

ラベル **lblAll** (レコード数)

→ 「/」と、名前「件数」のセルの値を表示する。

レコード番号は、表示データの行番号から定数 **S_ROW** を引いた値になる。

<コラム>ListIndex プロパティ

ListIndex プロパティは、リストボックスやコンボボックスで選択されている項目を上から 0、1、2……の数値で返します。何も選択されていない場合は ListIndex は -1 を返します。

ここでは、[区分] 列のセルが空欄の場合は、「リストボックスで何も選択しない」という意味で ListIndex に -1 を代入しています。

●フォームを開いたときに 1 レコード目を表示する

フォームを開いたときに表の 1 レコード目（1 件目）を表示するには、フォームの Initialize イベントプロシージャの最後に次のコードを追加します。表の 1 件目の顧客 No の値を引数に設定した、[レコード読み込み] プロシージャを呼び出します。

なお、データが 1 件も入力されていない場合は [顧客 No] に 1 を表示します。

【Sample】フォームを開くときに実行する Initialize イベントプロシージャ

```
Private Sub UserForm_Initialize()  
    Dim i As Long  
        : (省略)  
    If Range("A4").Value = "" Then  
        txtNo.Text = 1  
        Set rRng = Range("A4")  
    Else  
        Call レコード読み込み(Range("A4").Value)  
    End If  
End Sub
```

セル A4（1 レコード目のセル）に何も入力されていない場合

→ テキストボックス「txtNo」（顧客 No）に 1 を表示し、変数 rRng にセル A4 を代入する。

データが入力されていた場合

→ A4 の値を引数にして [レコード読み込み] プロシージャを呼び出す。

■ 1 件目のレコードが表示される

顧客NO	氏名	フリガナ	性別	区分	生年月日	年齢
1	正木 希海	マサキ ノノミ	女	レギュラー	1995/01/18	27
2	野村					44
3	西					41
4	久					62
5	田					33
6	西					36
7	松					22
8	河					47
9	田					
10	山					
11	橋					
12	松					38
13	松					44
14	梅					58
15	木					26
16	浜					67

lblAll.Caption : 全レコード数 : セル D1 (名前 [件数]) の値

lblRec.Caption : レコード番号 : セル A4 の行番号から定数 S_ROW (3) を引いた値になる

● [次] [前] ボタンでデータを切り替える

[次] ボタンでは、現在表示しているレコードの 1 行下のレコードを表示し、

[前] ボタンでは、1 行上のレコードを表示します。現在表示しているレコードが先頭の場合/最後の状態で処理を分けた上で引数 cNo の値を取得し、[レコード読み込み] プロシージャを呼び出します。

【Sample】 [次] ボタン (btnNext) がクリックされた際の処理

```
Private Sub btnNext_Click()  
    If rRng.Row = Range("A" & Rows.Count).End(xlUp).Row Then Exit Sub---①  
    Call レコード読み込み(rRng.Offset(1).Value)-----②  
End Sub
```

①現在表示しているデータの行番号が表の最下行のセルの行番号と同じ場合（最後の状態）は、処理を終了する。

②現在表示しているデータの 1 つ下の行のセルの値を引数にして [レコード読み込み] プロシージャを呼び出す。

■[次]ボタンクリック時

顧客NO	氏名	フリガナ	性別	区分	生年月日	年齢
1	正木 希瀬	マサキ ノゾミ	女	レギュラー	1995/01/18	27
2	野本 聡	ノモト サトシ	男	ゴールド	1977/10/24	44
3	西野 重子					
4	久野 正行					
5	茂木 里美					
6	西田 秀					
7	松本 明					
8	河合 夏樹					
9	田端 靖男					
10	山脇 歩美					
11	橋本 佳紀					
12	松本 敏次					
13	松井 悦浩					
14	梅原 幸之					
15	木下 優希					
16	浜田 健之					

顧客No
2

氏名
野本 聡

フリガナ表示
ノモト サトシ

性別
☒ 男 ☐ 女

区分
レギュラー

生年月日
1977 / 10 / 24

顧客NOで検索

先頭 前 次 最後 新規 2 / 20 登録 終了(O)

顧客マスター

準備完了 国 アクセシビリティ: 検封が必要です

[次] ボタンをクリックすると、次のレコードが表示される

【Sample】 [前] ボタン (btnPre) クリック時に実行するイベントプロシージャ

```
Private Sub btnPre_Click()  
    If rRng.Row = Range("A4").Row Then Exit Sub ----- ①  
    Call レコード読み込み(rRng.Offset(-1).Value) ----- ②  
End Sub
```

①現在表示しているデータの行番号がセル **A4** の行番号と同じ場合（先頭の場合）は、処理を終了する。

②現在表示しているデータの1つ上の行のセルの値を引数にして **[レコード読み込み]** プロシージャを呼び出す。

■[前]ボタンクリック時

雇入NO	氏名	フリガナ	性別	区分	生年月日	年齢
1	正木 希海	マサキ ノノミ	女	レギュラー	1995/01/18	27
2	野本 聡	ノモト サトシ	男	ゴールド	1977/10/24	44
3	西野 豊子	サイノノ ヒロコ				
4	久野 正行	クノ マサユキ				
5	茂木 里美	モリキ リミ				
6	西田 健	サイタ ケン				
7	松本 明	マツモト アキラ				
8	河合 夏希	カワイ ナツキ				
9	田端 清男	タナベ ヒロユキ				
10	山脇 歩美	ヤマワキ ヒロミ				
11	橋本 佳代	ハシモト ユイ				
12	松井 悦法	マツイ エツホウ				
13	梅原 幸之	ウメハラ ユキノブ				
14	木下 優子	キノシタ ユウコ				
15	美田 健之	ミタ ケンノブ				

● [最後] [先頭] ボタンでデータを切り替える

[最後] ボタンでは最終行のレコードを表示し、**[先頭]** ボタンでは先頭行のレコードを表示します。

最終レコードは表の一番下のセルの値、先頭レコードはセル **A4** の値を引数にして **[レコード読み込み]** プロシージャを呼び出します。

【Sample】 [最後] ボタン (btnLast) がクリックされた際の処理

```
Private Sub btnLast_Click()  
    Call レコード読み込み(Range("A" & Rows.Count).End(xlUp).Value)  
End Sub
```

セル **A** 列のワークシートの最下行から上端のセル（表の一番下のレコード）の値を引数に指定 [レコード読み込み] プロシージャを呼び出す。

■[最後]ボタンクリック時

[最後] ボタンをクリックすると、最終レコードが表示される

【Sample】 [先頭] ボタン (btnTop) がクリックされた際の処理

```
Private Sub btnTop_Click()  
    Call レコード読み込み(Range("A4").Value)  
End Sub
```

セル **A4** (先頭レコード) の値を引数にして [レコード読み込み] プロシージャを呼び出す。

■[先頭]ボタンクリック時

[先頭] ボタンをクリックすると、先頭レコードが表示される

● [新規] ボタンでコントロールを初期化する

[新規] ボタンを押したら、新規入力用にフォームに表示されているデータをクリアし、テキストボックス `txtNo` に最終レコードの顧客 NO に 1 を加算した値を設定します。

コントロールの値をリセットするために `For Each ステートメント` を使って `TypeName` 関数でコントロールの種類を調べ、コントロールの種類によってリセットの方法を指定しています。

【Sample】 [新規] ボタン (`btnNew`) がクリックされた際の処理

```
Private Sub btnNew_Click()  
    Dim lstRng As Range '表の最終レコードのセル代入用  
    Dim obj As Control  
    'A: コントロールの値をリセット  
    For Each obj In UserForm1.Controls ----- ①  
        Select Case TypeName(obj)  
            Case "TextBox"  
                obj.Value = ""  
            Case "ListBox", "ComboBox" ----- ②  
                obj.ListIndex = -1  
            Case "OptionButton"  
                obj.Value = False  
        End Select  
    Next  
    'B: 最終セルと新規データ用セルの取得  
    Set lstRng = Range("A" & Rows.Count).End(xlUp) ----- ③  
    Set rRng = lstRng.Offset(1) ----- ④  
    'ラベルに [新規レコード] と表示  
    lblRec.Caption = "" ----- ⑤  
    lblAll.Caption = "新規レコード"  
    txtNo.Value = lstRng.Value + 1 ----- ⑥  
End Sub
```

ブロック A:

コントロールの値を初期化します。

```
'A: コントロールの値をリセット
For Each obj In UserForm1.Controls -----①
    Select Case TypeName(obj)
        Case "TextBox"
            obj.Value = ""
        Case "ListBox", "ComboBox" -----②
            obj.ListIndex = -1
        Case "OptionButton"
            obj.Value = False
    End Select
Next
```

①オブジェクト型の変数 **obj** に **UserForm1** 上のすべてのコントロールを順番に代入しながら、以下の処理を実行する。

②変数 **obj** に代入されたコントロールの種類が

テキストボックス → **Value** プロパティに「**""**」を代入する。

リストまたはコンボボックス → **ListIndex** プロパティに「**-1**」を代入する。

オプションボタン → オフに設定する (**False**)。

ブロック B:

最終セルと新規データ用のセルを取得します。

```
'B: 最終セルと新規データ用セルの取得

Set lstRng = Range("A" & Rows.Count).End(xlUp)-----③

Set rRng = lstRng.Offset(1) ----- ④

'ラベルに [新規レコード] と表示

lblRec.Caption = "" ----- ⑤

lblAll.Caption = "新規レコード"

txtNo.Value = lstRng.Value + 1 -----⑥
```

③表の最後のレコードの **A** 列のセルを変数 **lstRng** に代入する。

④最後のレコードの 1 つ下 (新規レコード) のセルを変数 **rRng** に代入する。

⑤ラベル **lblRec** に何も表示せず、**lblAll** に [新規レコード] と表示する。

⑥**lstRng** (最終行の **A** 列のセル) の値に **1** を加算して顧客 No を採番し、**txtNo** に表示する。

■[新規]ボタンクリック時

最終レコードの No に 1 を加算した値が表示される

コントロールの値がリセットされる

ラベルに[新規レコード]と表示される

● [登録] ボタンでフォームの値をセルに転記する

[登録] ボタンをクリックしたら、**ユーザーフォームに表示されているデータを表に追加する**イベントプロシージャを作成します。登録するデータを確認し、新規レコードなら登録、既存のレコードなら上書きするかどうか確認メッセージを表示します。書き込み先として変数 **rRng** のセルを使います。モジュールレベルの変数 **rRng** には、フォームに表示されている「顧客 No」の値のセルへの参照が代入されています。

ここでは、各コントロールの値をセルの **Value** プロパティに代入できる値に直して、変数 **rRng** のセルの行の **1** 列目 (**A** 列) から **6** 列目 (**F** 列) に順番に書き出していきます。

【Sample】 [登録] ボタン (btnSave) がクリックされた際の処理

```
Private Sub btnSave_Click()  
    Dim obj As Control, r As Long  
    Dim birth As String, rng As Range  
    'A: データチェック  
    birth = cmbYear.Value & "/" & cmbMonth.Value & "/" & cmbDay.Value ----- ①  
    Select Case True ----- ②  
        Case txtNo.Value = ""  
            MsgBox "顧客NOを入力してください" ----- ③  
            Exit Sub  
        Case txtName.Value = ""  
            MsgBox "氏名を入力してください" ----- ④  
            Exit Sub  
        Case optMan.Value = False And optWoman.Value = False  
            MsgBox "性別を選択してください" ----- ⑤  
            Exit Sub  
        Case lstKubun.ListIndex = -1  
            MsgBox "区分を選択してください" ----- ⑥  
            Exit Sub  
        Case IsDate(birth) = False  
            MsgBox "正しい日付を入力してください" ----- ⑦  
            Exit Sub  
    End Select  
    'B: 重複チェック  
    Set rng = Range("A4").CurrentRegion.Columns(1).Find( _  
        What:=txtNo.Value, lookat:=xlWhole) ----- ⑧  
    '同じ顧客番号が見つかった場合  
    If Not rng Is Nothing Then  
        rng.Select ----- ⑨  
        If MsgBox("上書きしますか?", vbYesNo) = vbNo Then Exit Sub ----- ⑩  
        Set rRng = rng ----- ⑪  
    End If
```

'C: 書き込み

```
r = rRng.Row ----- ⑫  
Cells(r, 1).Value = txtNo.Value  
Cells(r, 2).Value = txtName.Value  
Cells(r, 3).Value = txtKana.Value  
If optMan.Value Then  
    Cells(r, 4).Value = "男" ⑭  
Else  
    Cells(r, 4).Value = "女"  
End If  
Cells(r, 5).Value = lstKubun.Value ----- ⑮  
Cells(r, 6).Value = birth ----- ⑯  
Cells(r, 6).NumberFormatLocal = "yyyy/mm/dd" ----- ⑰  
Cells(r, 7).Formula = "=IF(" & Cells(r, 6).Address & _  
    "=""", "", DATEDIF(" & Cells(r, 6).Address & _  
    ", TODAY(), "Y"))" ⑱  
'ラベルに現在のレコード番号とレコード数を表示  
lblRec.Caption = rRng.Row - S_ROW ----- ⑲  
lblAll.Caption = "/" & Range("件数").Value ----- ⑳  
End Sub
```

このコードは大きく 3 つのブロックに分かれています。

A: 入力欄の空欄チェック、データ型チェック

B: 入力内容の重複チェック

C: レコード書き込み

次ページではブロックごとにコードの解説を行います。

ブロック A:

入力欄が空欄かどうか、生年月日のコンボボックスに入力された値が日付として扱えるかどうかをチェックしています。

'A: データチェック

```
birth = cmbYear.Value & "/" & cmbMonth.Value & "/" & cmbDay.Value ---①
Select Case True ----②
    Case txtNo.Value = ""
        MsgBox "顧客 NO を入力してください" ③
        Exit Sub
    Case txtName.Value = ""
        MsgBox "氏名を入力してください" ④
        Exit Sub
    Case optMan.Value = False And optWoman.Value = False
        MsgBox "性別を選択してください" ⑤
        Exit Sub
    Case lstKubun.ListIndex = -1
        MsgBox "区分を選択してください" ⑥
        Exit Sub
    Case IsDate(birth) = False
        MsgBox "正しい日付を入力してください" ⑦
        Exit Sub
End Select
```

① コンボボックス **cmbYear**、**cmbMonth**、**cmbDay** の値を「/」でつなげて「2018/11/26」の形式で文字列にして変数 **birth** に代入する。

② 以下の値が **True** かどうか判定する。

③ テキストボックス **txtNo** の値が「」の場合、

「顧客 NO を入力してください」とメッセージを表示して処理を終了する。

④ テキストボックス **txtName** の値が「」の場合、

「氏名を入力してください」とメッセージを表示して処理を終了する。

⑤ オプションボタン **optMan** と **optWoman** の値がともに **False** の場合、

「性別を選択してください」とメッセージを表示して処理を終了する。

⑥ リストボックス **lstKubun** が何も選択されていない場合、

「区分を選択してください」とメッセージを表示して処理を終了する。

⑦ 変数 **birth** が日付とみなされない場合、

「正しい日付を入力してください」とメッセージを表示して処理を終了する。

<Memo>

IsDate関数は、引数に指定した値が日付や時刻として扱えるかどうかを調べて扱える場合はTrue、扱えない場合はFalseが返ります（p.358参照）

ブロック B:

テキストボックス **txtNo** の値が既に入力済みのレコードと重複するかどうかをチェックしています。

```
'B: 重複チェック
Set rng = Range("A4").CurrentRegion.Columns(1).Find( _
    What:=txtNo.Value, lookat:=xlWhole) ⑧
'同じ顧客番号が見つかった場合
If Not rng Is Nothing Then
    rng.Select ---- ⑨
    If MsgBox("上書きしますか?", vbYesNo) = vbNo Then Exit Sub --⑩
    Set rRng = rng ---- ⑪
End If
```

⑧セル **A4**を含む表の1列目でテキストボックス **txtNo** の値（顧客 No）を検索し、見つかったら変数 **rng** に代入する。

⑨見つかったセルを選択する。

⑩メッセージに対して[No]ボタンが選択された場合、処理を終了する。

⑪変数 **rRng** に変数 **rng** の値を代入する。
（見つかったセルが書き換え対象のセルとなる）

ブロック C:

ユーザーフォームに表示されている値を[顧客マスター]表に書き込みます。

'C: 書き込み

```

r = rRng.Row ----- ⑫
Cells(r, 1).Value = txtNo.Value
Cells(r, 2).Value = txtName.Value
Cells(r, 3).Value = txtKana.Value
If optMan.Value Then
    Cells(r, 4).Value = "男"
Else
    Cells(r, 4).Value = "女"
End If
Cells(r, 5).Value = lstKubun.Value --- ⑮
Cells(r, 6).Value = birth ----- ⑯
Cells(r, 6).NumberFormatLocal = "yyyy/mm/dd" --- ⑰
Cells(r, 7).Formula = "=IF(" & Cells(r, 6).Address & _
    "=""", , """, DATEDIF(" & Cells(r, 6).Address & _
    ",TODAY(),"Y"))"
'ラベルに現在のレコード番号とレコード数を表示
lblRec.Caption = rRng.Row - S_ROW ----- ⑰
lblAll.Caption = "/" & Range("件数").Value ----- ⑳
```

⑫変数 **rRng** の行番号を変数 **r** に代入する。

⑬テキストボックスの値をそれぞれ、変数で指定したセルに入力する。

txtNo→**r**行1列目、**txtName**→**r**行2列目、**txtKana** → **r**行3列目

⑭オプションボタン **optMan** の値が

True の場合 → **r**行4列目に「男」と表示する。

そうでない場合 → **r**行4列目に「女」と表示する。

⑮リストボックス **lstKubun** で選択されている値を **r**行5列目に表示する。

⑯変数 **birth** の値を **r**行6列目に表示する。

⑰**r**行6列目のセルの表示形式を「**yyyy/mm/dd**」に設定する。

⑱**r**行7列目のセルに **DATEDIF関数**と **IF関数**と組み合わせて

「**=IF(\$F\$23="", "", DATEDIF(\$F\$23, TODAY(), "Y"))**」となるように、年齢を求める数式を入力する。

- ①9 変数 **rRng** の行番号から定数 **S_ROW** を引いてレコード番号を求め、ラベル **lblRec** に表示する。②0 [/] と名前 [件数] のセルの値をラベル **lblAll** に表示する（これで [レコード番号/全レコード数] が表示される）。

<コラム> DATEDIF 関数で年齢を求める式をセルに入力する

ワークシート関数の **DATEDIF 関数** を使って年齢を求める式を入力する場合、以下の書式で設定します。

<書式> DATEDIF関数

=DATEDIF(生年月日, TODAY(), "Y")

第1引数 : 生年月日

第2引数 : 今日の日付を求めるTODAY関数

第3引数 : 単位「"Y"」

第1引数でセルを参照する場合は、Range オブジェクトの Address プロパティを使ってセル番地を取得します。Address プロパティに引数を指定しないと絶対参照となります（DATEDIF関数はP.352、AddressプロパティはP.311を参照）。

サンプルでは生年月日が空欄の場合を考慮し、If 関数と組み合わせています。

■新規レコードの場合

■既存レコードの場合

既存の顧客 NO の場合は、そのレコードのセルを選択し、上書きを確認するメッセージを表示する
[はい] → そのレコードに上書き [いいえ] → 何もせず終了

<Memo>

既存のレコードで [顧客 NO] を変更し、重複がなければそのまま書き込まれます。

- 「顧客 NO で検索」ボタンをクリックするとテキストボックスに入力された顧客 NO を持つレコードを検索し、見つかったらユーザーフォームに読み込みます。

【Sample】 [顧客 NO で検索] ボタン (btnKensaku) がクリックされた際の処理

①テキストボックス `txtKensaku` に入力された値が数値として扱えない場合は「顧客NOを入力してください」とメッセージを表示して処理を終了する。

②`txtKensaku` に入力された値を引数にして「レコード読み込み」プロシージャを呼び出す。

30

<コラム> ユーザーフォームに表示されているデータを削除する

ユーザーフォームに表示されているデータを削除するコードを紹介します。

ユーザーフォームに [削除] ボタン (btnDel) を追加し、「btnDel_Click」イベントプロシージャでコードを記述します。

ここでは、削除するデータが最終レコードかどうかを確認し、最終レコードを削除した場合は削除後にユーザーフォームに 1 つ上のレコードを表示し、最終でない場合は 1 つ下のレコードを表示するように引数を取得し、「レコード読み込み」を呼び出します。

【Sample】 [削除] ボタン (btnDel) がクリックされた際の処理

```
Private Sub btnDel_Click()  
    Dim ans As Integer      'メッセージの戻り値用  
    Dim no As Long         '引数となる顧客NO代入用  
    ans = MsgBox("削除すると元に戻せません" & vbCrLf & "よろしいですか?", ①  
        vbYesNo + vbExclamation, "削除確認")  
    If ans = vbNo Then Exit Sub ----- ②  
    If rRng.Row = Range("A" & Rows.Count).End(xlUp).Row Then  
        no = rRng.Offset(-1).Value  
    Else  
        no = rRng.Offset(1).Value  
    End If  
    Range(rRng, rRng.Offset(, 6)).Delete xlShiftUp ----- ④  
    Call レコード読み込み(no) ----- ⑤  
End Sub
```

① [削除すると元に戻せません] [よろしいですか?] とメッセージを改行表示する。

② メッセージで [いいえ] ボタンが選択された場合、処理を終了する。

③ 変数 rRng (ユーザーフォームに表示しているデータのセル) の行番号が最終レコードの行と同じ場合→変数 no に変数 rRng の 1 つ上のセルの値を代入する。そうでない場合→変数 no に変数 rRng の 1 つ下のセルの値を代入する。

④ 変数 rRng のセルからそのセルの 6 列右側のセルまでを削除し、削除後、下の行を上シフトする。

⑤ 引数に変数 no を持たせて [レコード読み込み] プロシージャを呼び出す。

①[顧客 No12][松本敏次]が表示された状態で[削除]ボタンをクリックする

②確認メッセージ画面で[はい]を選択する

③表示されていたデータが削除され、1つ下のレコードがフォームに表示される

顧客NO	氏名	フリガナ	性別	区分	生年月日	年齢
12	松本 敏次	マツモト トシツグ	男	プレミアム	1983/12/17	38
13	松井 悦治	マツイ エツジ	男	プレミアム	1978/05/20	44
14	梅原 幸之助	ウメハラ コウノス	男	ゴールド	1963/11/05	58
15	木下 優奈	キノシタ ユウナ	女	ゴールド	1996/06/12	26
16	浜田 健之	ハマダ タケユキ	男	ゴールド	1955/03/27	67

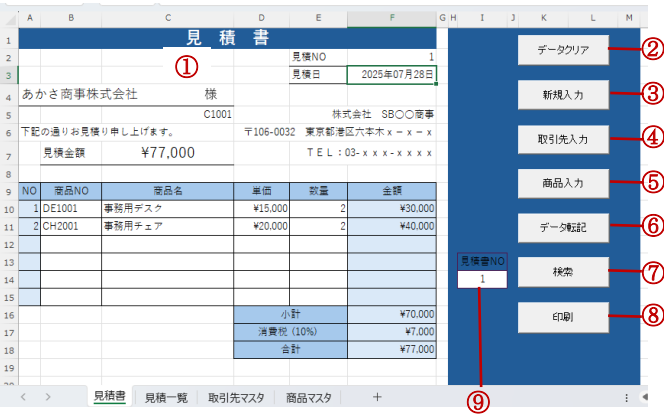
□02 簡易見積書作成システム

Sample_Data/ap-02-01.xlsm

● 簡易見積書作成システムの概要と各シートの構成

本章では、見積書のような明細行のある表への効率的なデータ入力、一覧へのデータ転記や読み込みなどの処理を、Excel の機能と VBA の機能を組み合わせた簡易見積書作成システムを紹介します。まずは、これから作成するシステムの概要と各シートの構成を確認してください。

■[見積書]シート



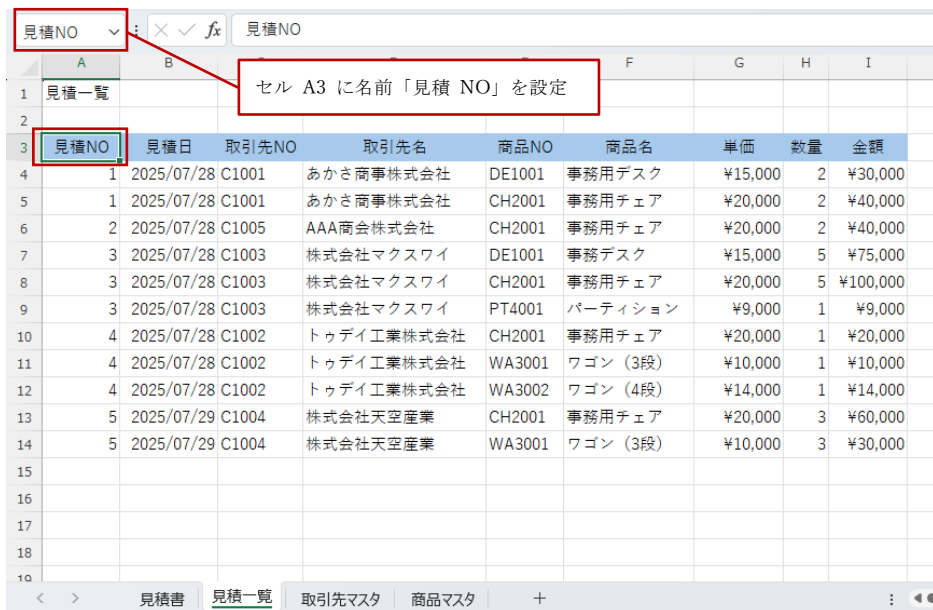
右側に配置したボタンをクリックして各処理を行います。

■[見積書]シートの項目

No	構成	内容
1	見積書	見積書のデータの入力、表示、印刷用。
2	[データクリア]ボタン	[データクリア]プロシージャ。見積NO、見積日、取引先名、取引先NO、明細データを消去する。
3	[新規入力]ボタン	[新規入力]プロシージャ。見積NO の採番と見積日を入力する。
4	[取引先入力]ボタン	ユーザーフォーム[取引先]を表示し、取引先、取引先NO を入力する。

No	構成	内容
5	[商品入力]ボタン	ユーザーフォーム[商品]を表示し、商品NO、商品名、単価を入力する。
6	[データ転記]ボタン	見積書の内容を[見積一覧]シートの一覧に転記する。
7	[検索]ボタン	名前[検索NO]のセルに入力された見積NOを[見積一覧]シートの一覧から検索し、 [見積書]シートに読み込む。
8	[印刷]ボタン	印刷設定をして、印刷を実行する。
9	名前[検索 NO]のセル	セルI14に名前[検索NO]を設定し、検索用の見積書NOを入力する。

■[見積一覧]シート



見積NO	:	×	✓	fx	見積NO					
A	B				F	G	H	I		
1	見積一覧									
2										
3	見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額	
4	1	2025/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000	
5	1	2025/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000	
6	2	2025/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000	
7	3	2025/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000	
8	3	2025/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000	
9	3	2025/07/28	C1003	株式会社マクスワイ	PT4001	パーティション	¥9,000	1	¥9,000	
10	4	2025/07/28	C1002	トゥデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000	
11	4	2025/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000	
12	4	2025/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000	
13	5	2025/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000	
14	5	2025/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000	
15										
16										
17										
18										
19										
20										

[データ転記] ボタン ([見積書]シートの⑥) をクリックすると、見積書の内容が[見積一覧]シートに書き出されます。

■[取引先マスタ]シート

	A	B	C	D
1	取引先マスター			
2				
3	取引先NO	会社名	担当者	
4	C1001	あかさ商事株式会社	田中 太郎	
5	C1002	トゥデイ工業株式会社	後藤 慎一郎	
6	C1003	株式会社マクスワイ	山下 紀子	
7	C1004	株式会社天空産業	鈴木 徹	
8	C1005	AAA商会株式会社	斉藤 浩二	
9				
10				
	<	見積書	見積一覧	取引先マスタ

セル A4~B8 に名前「取引先リスト」を設定

テーブル形式の取引先の一覧表です。セル範囲 **A4~B8** の値を **[取引先入力]** ボタン（[見積書]シートの④）をクリックしたときに開くユーザーフォームのリストボックスに表示します。テーブルにしているため、名前「取引先リスト」のセル範囲はデータの増減に対応できます。

■[商品マスタ]シート

	A	B	C	D
1	商品マスター			
2				
3	商品NO	商品名	単価	
4	DE1001	事務用デスク	¥15,000	
5	DE1002	会議用デスク	¥12,000	
6	DE1003	サイドデスク	¥8,000	
7	CH2001	事務用チェア	¥20,000	
8	CH2002	会議用チェア	¥13,000	
9	WA3001	ワゴン（3段）	¥10,000	
10	WA3002	ワゴン（4段）	¥14,000	
11	PT4001	パーティション	¥9,000	
12				
18				
10	<	見積書	見積一覧	取引先マスタ

セル A4~C11 に名前「商品リスト」を設定

テーブル形式の商品の一覧表です。セル範囲 **A4~C11** の値を **[商品入力]** ボタン（[見積書]シートの⑤）をクリックしたときに表示するユーザーフォームのリストボックスに表示します。テーブルにしているため、名前「商品リスト」のセル範囲は、データの増減に対応できます。

● 見積書に設定されている計算式

【見積書】シートには、次のような計算式が設定されています。見積書に商品NO や数量などのデータが入力されたら自動的に表示・計算されます。

The screenshot shows the '見積書' (Estimate) sheet with the following data and formulas:

No	商品NO	商品名	単価	数量	金額
1	CH2001	事務用デスク	¥15,000	2	¥30,000
2	CH2001	事務用チェア	¥20,000	2	¥40,000
小計					¥70,000
消費税 (10%)					¥7,000
合計					¥77,000

Formulas shown in callouts:

- ①=F18 (Cell F18)
- ②=IF (B10="", "", ROW() - 9) (Cell A10)
- ③=IF (B10="", "", D10*E10) (Cell F10)
- ④=SUM (F10:F15) (Cell F16)
- ⑤=F16*E17 (Cell F17)
- ⑥=SUM (F16:F17) (Cell F18)

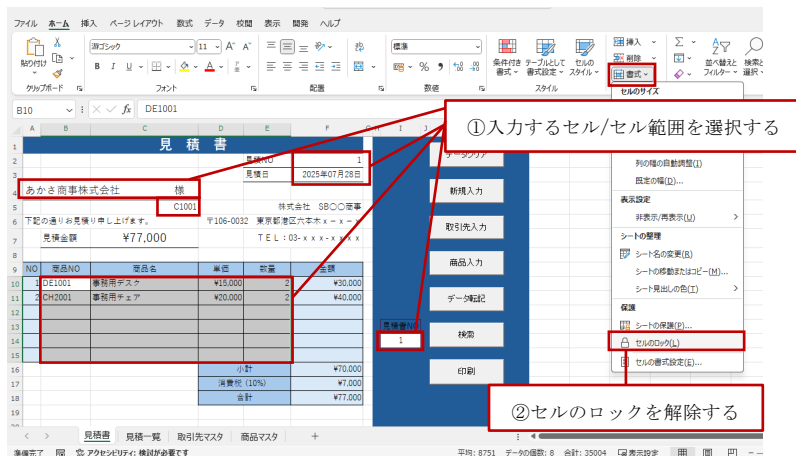
No	セル	式	内容
1	C7	=F18	合計のセル F18 を参照する。
2	A10	=IF (B10="", "", ROW() - 9)	明細行の連番を自動表示する。 セル B10 が空欄でない場合に現在の行番号から 9を引いた数を表示し、セル A15 までコピーする。
3	F10	=IF (B10="", "", D10*E10)	単価×数量。商品 NO が空欄の場合は表示しない。セル F15 までコピーする。
4	F16	=SUM (F10:F15)	明細行の金額を合計する。
5	F17	=F16*E17	小計×消費税率。消費税額を表示。セル E17 に消費税率「 10% 」が入力されている。
6	F18	=SUM (F16:F17)	小計+消費税額。見積金額を表示する。

<コラム>[シートの保護]で計算式を保護する

見積書の計算式が削除されないように、入力するセル以外は編集ができないように設定しておくで安心です。

セルは既定値でロックがオンになっているので、入力欄となるセルのロックを解除し、次にシートを保存します。手順は次の通りです。

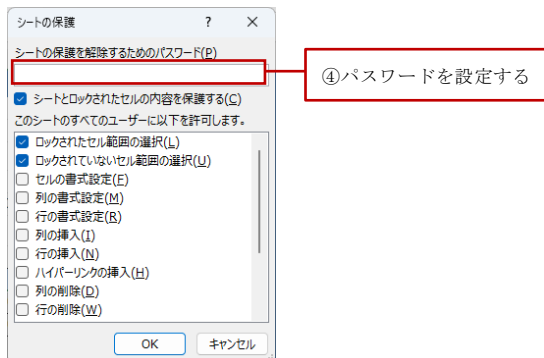
- ①入力に使うセル/セル範囲を選択する（ここでは F2~F3、A4、C5、B10~E15、I14）。
- ②[ホーム]タブの[書式]→[セルのロック]を選択し、選択したセルのロックを解除する。



- ③[校閲]タブの[シートの保護]をクリックする。



- ④[シートの保護]ダイアログで必要に応じてパスワードを設定し、[OK]で確定する。

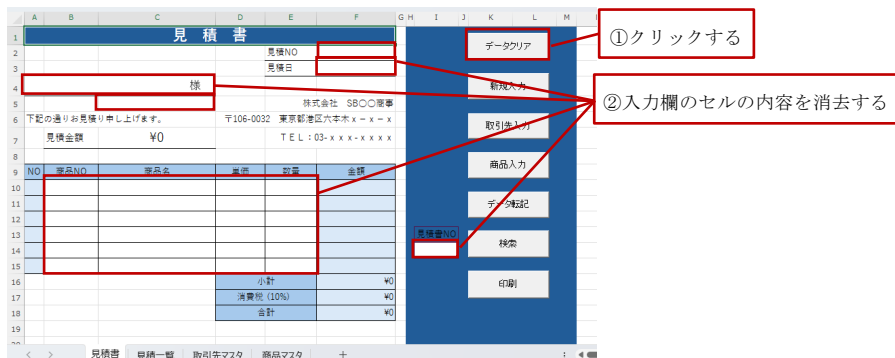


また、シートの保護を解除するには、[校閲]タブの[シート保護の解除]をクリックします。パスワードを設定した場合は、パスワード入力画面が表示されるので設定したパスワードを入力します。パスワードを忘れてしまうと解除できなくなるので気を付けてください。

- [データクリア]ボタンで入力欄のデータを消去する

[データクリア] ボタンには [データクリア] プロシージャを割り当てます。[データクリア] ボタンをクリックすると、**入力欄となるセル（セル F2、F3、A4、C5、B10～E15、I14）の内容を消去**します。

なお、セル A4はセル A4～C4が結合されており、表示されている「様」は、図形のテキストボックスを配置しているため、消去されません。



[データクリア] プロシージャは次ページのようになります。入力欄のセルに対して **ClearContents メソッド** でセルの内容を消去します。セル A4 は結合セルなので、**MergeAreaプロパティ** を使って結合セルを参照しています。

【Sample】 [データクリア] プロシージャ

```
Sub データクリア ()
    With Worksheets("見積書")
        Range("F2:F3").ClearContents
        Range("A4").MergeArea.ClearContents
        Range("C5").ClearContents
        Range("B10:E15").ClearContents
        Range("検索 NO").ClearContents
    End With
End Sub
```

<Memo>

結合セルを参照する場合、MergeAreaプロパティを使う以外に、結合されているすべてのセルを参照しても処理できます。ここではセルA4～C4が結合されているので、「Range("A4:C4").ClearContents」と記述できます。

● [新規入力]ボタンで新規入力の準備をする

[新規入力] ボタンには [新規入力] プロシージャを割り当てます。[新規入力] ボタンをクリックすると、[データクリア] プロシージャを呼び出して入力欄のデータを消去し、見積 NO を採番してセル F2 に入力し、セル F3 の見積日に今日の日付を入力します。

【Sample】 [新規入力] プロシージャ

```
Sub 新規入力 ()  
    Dim newID As Long  
    Call データクリア  
    newID = Application.WorksheetFunction.Max( _  
        Range("見積 NO").CurrentRegion.Columns(1)) + 1  
    Range("F2").Value = newID  
    Range("F3").Value = Date  
End Sub
```

①
②
③

①MAX 関数を使って名前「見積 NO」のセルを含む表全体の 1 列目の中から最大値を求め、1 を加算して変数newID に代入する。

→変数newID が採番された見積 NOになる。

②セルF2に変数newID の値を表示する。

③セルF3にDate 関数で今日の日付を表示する。

<Memo>

「見積一覧」シートのセルA3に名前「見積 NO」が設定されています。

■採番の仕組み

見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
1	2025/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
2	2025/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2025/07/28	C1005	AAA商會株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
4	2025/07/28	C1003	株式会社マクスウィ	DE1001	事務用デスク	¥15,000	5	¥75,000
5	2025/07/28	C1003	株式会社マクスウィ	CH2001	事務用チェア	¥20,000	5	¥100,000
6	2025/07/28	C1003	株式会社マクスウィ	PT4001	パーティション	¥9,000	1	¥9,000
7	2025/07/28	C1002	トゥデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000
8	2025/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
9	2025/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
10	2025/07/28	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
11	2025/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

Application.WorksheetFunction.Max(Range("見積 NO").CurrentRegion.Columns(1)) + 1
 名前が「見積 No」のセルを含む表全体の 1 列目の中で最大値を求め、1 を加算して見積番号を採番する

●[取引先入力]ボタンで取引先を一覧から選択・入力する

「取引先入力」ボタンをクリックすると、取引先 NO と取引先名が一覧表示されているユーザーフォームを表示し、そこから追加する取引先を選択して「入力」ボタンをクリックすると見積表のセルに表示する仕組みを作ります。

The screenshot shows a spreadsheet with a table titled '見積書' (Estimate). The table has columns for 'NO', '商品NO', and '商品名'. A dialog box titled '取引先入力' (Vendor Input) is open, showing a list of vendors. The first vendor, 'あかさ商事株式会社' (Akasaka Shoji Co., Ltd.), is selected. The '入力' (Input) button is highlighted in the dialog box.

①クリックする

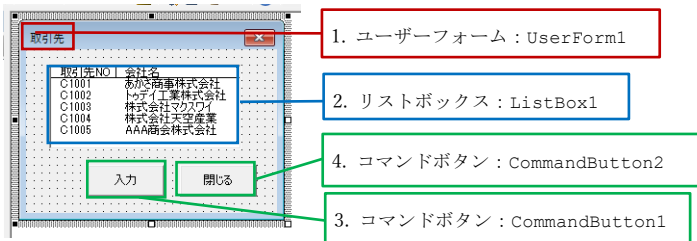
②フォームの一覧から取引先を選択し、「入力」をクリックする

③取引先名 (A4) と取引先 NO (C5) が見積書に入力される

【Sample】 「取引先入力」 ボタンに割り当てられているプロシージャ

```
Sub 取引先フォーム表示 ()
    UserForm1.Show          'UserForm1 を開く
End Sub
```

「取引先」 フォームは、リストボックスとコマンドボタンを以下のように配置して作成します。



■各コントロールの[プロパティウィンドウ]での設定内容

ユーザーフォーム/コントロール名	プロパティ	設定値
1. ユーザーフォーム	(オブジェクト名)	UserForm1
	Caption	取引先
2. リストボックス	(オブジェクト名)	ListBox1
	ColumnCount	2
	ColumnHeads	True
	ColumnWidths	55pt; 70pt
	RowSource	取引先リスト
3. コマンドボタン	(オブジェクト名)	CommandButton1
	Caption	入力
4. コマンドボタン	(オブジェクト名)	CommandButton2
	Caption	閉じる

<コラム>リストボックスの設定

ここでは、リストボックスを2列に設定し、選択肢には、名前「取引先リスト」が付いているセル範囲を指定しています。

リストボックスを2列に設定するには ColumnCount プロパティで列数を「2」に指定し、列見出しを表示するには ColumnsHeads を True にします。ColumnWidths プロパティで1列目と2列目をセミicolon「;」で区切って「55pt;70pt」のようにポイント単位で指定します。

ユーザーフォームでリストボックスの値を選択して **【入力】ボタン (CommandButton1)** をクリックしたときに、**取引先 NO をセルC5、取引先名をセルA4 に表示** したいです。そのため、リストボックスで選択された行の**1**列目、**2**列目の値をそれぞれセル**C5、A4** の **Value** プロパティに代入します。

リストボックスで選択された行は **ListIndex** プロパティで取得できます。複数列の場合は、**List** プロパティでリストボックス内の行と列を指定して取得します。行、列ともに **1** 行目、**1** 列目は**0**から順番に数えます。

<書式> List プロパティの構文

ListBox オブジェクト.List (Row, Column)

引数 Row: リストの行を、先頭行を 0 として上から何番目か数値で指定
リストボックスの ListIndex プロパティで取得した値を指定できる

引数 Column: リストの列を、先頭列を 0 として左から何番目か数値で指定

【Sample】【入力】ボタン (CommandButton1) クリック時のイベントプロシージャ

```
Private Sub CommandButton1_Click()  
    Range("A4").Value = ListBox1.List(ListBox1.ListIndex, 1) -----①  
    Range("C5").Value = ListBox1.List(ListBox1.ListIndex, 0) -----②  
End Sub
```

① リストボックスで選択された行の**2**列目の値をセル**A4**に表示する。

② リストボックスで選択された行の**1**列目の値をセル**C5**に表示する。

セル**A4**は結合セルですが、データを入力する場合は、結合セル内の先頭のセルを指定するだけで設定できます。

【Sample】 [閉じる] ボタン (CommandButton2) クリック時のイベントプロシーダ

```
Private Sub CommandButton2_Click()  
    Unload Me  
End Sub
```

■ リストボックスの列の値を取得

取引先

ListIndex1.List(ListBox1.Listindex,0)

ListIndex1.List(ListBox1.Listindex,1)

入力 閉じる

● [商品入力]ボタンで商品を一覧から複数選択して入力

[商品入力] ボタンをクリックすると、商品NO、商品名、単価が一覧表示されているユーザーフォームを表示し、追加する商品を複数選択して [入力] ボタンをクリックすると、見積書の明細行のセルに表示する仕組みを作ります。

② フォームの一覧から商品を選択し、[入力]をクリックする

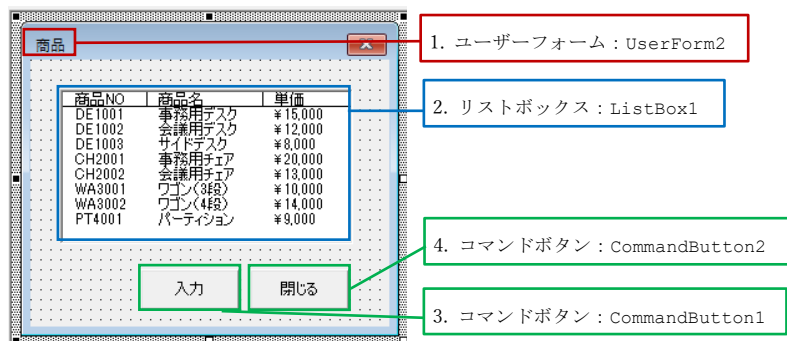
① クリックする

③ 商品情報が見積書の明細行に入力される

【Sample】 [商品入力] ボタンに割り当てられているプロシーダ

```
Sub 商品フォーム表示()  
    UserForm2.Show 'UserForm2 を開く  
End Sub
```

【商品】フォームは、リストボックスとコマンドボタンを以下のように配置して作成します。リストボックスは、名前「商品リスト」のセル範囲を選択肢として表示し、3列表示するため **ColumnCountプロパティ**を3に設定し、列見出しを表示するので **ColumnHeadsプロパティ**を**True**にします。また、ここでは複数選択できるように **MultiSelectプロパティ**を**True**にします。



■各コントロールの[プロパティウィンドウ]での設定内容

ユーザーフォーム/コントロール名	プロパティ	設定値
1. ユーザーフォーム	(オブジェクト名)	UserForm2
	Caption	商品
2. リストボックス	(オブジェクト名)	ListBox1
	ColumnCount	3
	ColumnHeads	True
	ColumnWidths	55 pt;70 pt;55 pt
	MultiSelect	True
	RowSource	商品リスト
3. コマンドボタン	(オブジェクト名)	CommandButton1
	Caption	入力
4. コマンドボタン	(オブジェクト名)	CommandButton2
	Caption	閉じる

ユーザーフォームでリストボックスの値を複数選択し、**【入力】ボタン (CommandButton1)** をクリックしたときに明細行に商品 NO、商品名、単価が入力されるようにするには、**List プロパティ**と **ListIndex プロパティ**を使ってリストボックスの列の値を取得します。

ここでは、選択された複数の商品を明細行に追加します。

リストボックスの項目が選択されているかどうかを、1つ目 (0) から、最後の項目 (**ListBox1.ListCount-1**) まで**繰り返し処理**で確認し、選択されていた場合に書き込みます。

また、選択されているかどうかは、リストボックスの **Selected プロパティ**を使って確認します。例えば、「**ListBox1.Selected(0)**」が **True** の場合、1 つ目の項目が選択されています。

【Sample】 [入力] ボタン (CommandButton1) クリック時のイベントプロシージャ

```
Private Sub CommandButton1_Click()  
    Dim meisaiCell As Range '明細行の先頭のセル代入用  
    Dim i As Integer        '繰り返し処理用  
    Range("B10:E15").ClearContents '明細範囲クリア  
    Set meisaiCell = Range("B10") ----- ①  
    For i = 0 To ListBox1.ListCount - 1 ----- ②  
        If ListBox1.Selected(i) Then ----- ③  
            meisaiCell.Value = ListBox1.List(i, 0)  
            meisaiCell.Offset(, 1).Value = ListBox1.List(i, 1) ④  
            meisaiCell.Offset(, 2).Value = ListBox1.List(i, 2)  
            Set meisaiCell = meisaiCell.Offset(1)----- ⑤  
        End If  
    Next  
    Set meisaiCell = Nothing 'オブジェクト変数の解放  
End Sub
```

①セル **B10** (明細行の先頭) を変数 **meisaiCell** に代入する。

②変数 **i** が **0** からリストボックスの項目数**-1** の数になるまで

以下の処理を繰り返す (リストボックスの項目は上から **0, 1, 2...**と数えるため **0** から始まり、項目数から **1** を引いた数で全項目を参照することになる)。

③リストボックスの **i** 番目の項目が選択されている場合、以下の処理を実行する。

④リストボックスの

i+1行目 **1**列目の値を変数 **meisaiCell** のセルに入力する。

i+1行目 2列目の値を変数 `meisaiCell` から 1つ右のセルに入力する。

i+1行目 3列目の値を変数 `meisaiCell` から 2つ右のセルに入力する。

- ⑤変数 `meisaiCell` の 1つ下のセルを変数 `meisaiCell` に設定し直し、繰り返しの処理 (②) に戻る。

<Memo>

「i+1 行目」について、リストボックスの項目は1番上が0であり、変数 `i` は0から数えているので、1を加算して実際の行数に直して説明しています。

【Sample】 [閉じる] ボタン (CommandButton2) クリック時のイベントプロシージャ

```
Private Sub CommandButton2_Click()  
    Range("E10").Select      '数値入力セルを選択  
    Unload Me  
End Sub
```

<コラム> ListCountプロパティと ListIndexプロパティ

リストボックスの **ListCount プロパティ** はリストボックス内の項目数を返します。実際の数を返すので、3項目あれば3が返ります。取得のみです。

一方、**ListIndex プロパティ** は、現在選択されている項目を取得します。

ListIndex の値が 0 の時は 1 項目、1 の時は 2 項目が選択されています。何も選択されていない場合は、-1 が返ります。取得と設定ができます。

<コラム>リストボックスの Selectedプロパティ

リストボックスの **Selected プロパティ** は、引数で指定したインデックス番号の項目が選択されていれば True、選択されていなければ False が返ります。例えば、1 つ目の項目が選択されているかどうかは「**ListBox1.Selected(0)**」で調べられます。

● [データ転記]ボタンで見積書のデータを一覧に書き込む

[データ転記] ボタンをクリックすると、**[見積書]**シートの入力データチェック、見積 NO の重複チェックを行った後、**[見積一覧]**シートに転記する仕組みを作ります。



② 書き込み

見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
1	2025/07/28	C1001	あかき商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
1	2025/07/28	C1001	あかき商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
2	2025/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2025/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000
3	2025/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
3	2025/07/28	C1003	株式会社マクスワイ	PT4001	パーティション	¥9,000	1	¥9,000
4	2025/07/28	C1002	トゥデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000
4	2025/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
4	2025/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
5	2025/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
5	2025/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

今回作成する「データ転記」プロシージャの処理内容は

- ①入力データのチェック
- ②見積 NO の重複チェック
- ③データ転記

の3パートに分かれています。それぞれについて見ていきましょう

見積書に記入された内容を見積一覧に転記する前に、**入力欄に空欄があるかどうかを確認する**ためのコードが必要です。ここでは、「データ転記」プロシージャにおける、「入力データのチェック」を紹介します。

【Sample】「データ転記」プロシージャ ①入力データのチェック

```
Sub データ転記 ()
'その① 空欄チェック *****
    Dim meisaiRng As Range '明細入力範囲代入用 その①、③で使用
    Dim cnt As Long       '明細件数代入用 その①、③で使用
    Dim ary As Variant    '繰り返し処理用
    Dim rArray As Variant
    Dim rng As Range

'A:見積 NO～取引先 NO の空欄チェック
    rArray = Array(Range("F2"), Range("F3"), Range("A4"), Range("C5")) -- ①
    For Each ary In rArray
        If ary.Value = "" Then
            ary.Select
            MsgBox "未入力箇所があります。", vbExclamation, "入力チェック" ②
            Exit Sub
        End If
    Next

'B:明細行の空欄チェック
    Set meisaiRng = Range("B10:E15") ----- ③
    cnt = Application.WorksheetFunction.CountA(meisaiRng.Columns(1)) -- ④
    'データが1件も入力されていない場合
    If cnt = 0 Then
        Range("B10").Select
        MsgBox "明細行が入力されていません", vbExclamation, "明細チェック" ⑤
        Exit Sub
    End If
End Sub
```

```

Else
    ' データが入力されていた場合、空欄が1つでもあったらキャンセル
    Set meisaiRng = meisaiRng.Resize(cnt) -----⑥
    For Each rng In meisaiRng ----- ⑦
        If rng.Value = "" Then
            rng.Select
            MsgBox "明細行に空欄があります", vbExclamation, "明細チェック" ----- ⑧
            Exit For
        End If
    Next
End If

MsgBox "入力漏れチェック完了" ----- ⑨

' その② 見積 No 重複チェック*****
' その③ データ転記 *****
End Sub

```

ブロック A :

見積 NO、日付、取引先、取引先 NO のセルの空欄をチェックします。

```

'A: 見積 NO～取引先 NO の空欄チェック

rArray = Array(Range("F2"), Range("F3"), Range("A4"), Range("C5")) ----- ①
For Each ary In rArray
    If ary.Value = "" Then
        ary.Select
        MsgBox "未入力箇所があります。", vbExclamation, "入力チェック" ----- ②
        Exit Sub
    End If
Next

```

- ① **Array** 関数を使ってセル **F2**、**F3**、**A4**、**C5** を配列変数にし、**Variant**型の配列変数 **rArray** に代入する。
- ② 配列変数 **rArray** の各要素を順番に変数 **ary** に代入しながら次の処理を実行する。
変数 **ary** の値が空欄の場合
→変数 **ary** を選択し、「未入力箇所があります」とメッセージを表示して処理を終了する。

ブロック B :

明細行のセルの空欄をチェックします。

```
'B:明細行の空欄チェック

Set meisaiRng = Range("B10:E15") ----③

cnt = Application.WorksheetFunction.CountA(meisaiRng.Columns(1)) ---- ④

'データが1件も入力されていない場合

If cnt = 0 Then

    Range("B10").Select

    MsgBox "明細行が入力されていません", vbExclamation, "明細チェック" ⑤

    Exit Sub

Else

    'データが入力されていた場合、空欄が1つでもあったらキャンセル

    Set meisaiRng = meisaiRng.Resize(cnt)----⑥

    For Each rng In meisaiRng ---- ⑦

        If rng.Value = "" Then

            rng.Select

            MsgBox "明細行に空欄があります", vbExclamation, "明細チェック" ⑧

            Exit For

        End If

    Next

End If

MsgBox "入力漏れチェック完了" ---- ⑨
```

- ③セル範囲 **B10** から **E15** を変数 **meisaiRng** に代入する。
- ④変数 **meisaiRng** の 1 列目のデータ数を **COUNTA 関数** で取得し、変数 **cnt** に代入する。
- ⑤変数 **cnt** の値が **0** の場合（明細行にデータが入力されていない場合）
 - セル **B10** を選択し、「明細行が入力されていません」とメッセージを表示して処理を終了する。
- ⑥0 でない場合（明細行にデータが入力されている場合）
 - 変数 **meisaiRng** のセル範囲の行数を、データ件数分にサイズ変更する。
- ⑦変数 **meisaiRng** の各セルを変数 **rng** に代入しながら次の処理を繰り返す。
- ⑧セル **rng** の値が空欄の場合
 - セル **rng** を選択し、「明細行に空欄があります」とメッセージを表示して処理を終了する。

⑨処理が終了したことを確認するためにメッセージを表示する（完成時、この行は削除する）。

①セル F2、F3、A4、C5 を配列にして rArray に代入する

②For Each ステートメントで空欄かどうかチェックする（空欄の場合、メッセージを表示して処理を終了する）

	A	B	C	D	E	F	G
1	見積書						
2					見積NO	5	
3					見積日	2025年07月29日	
4	株式会社天空産業		様				
5			C1004		株式会社	SB〇〇商事	
6	下記の通り見積り申し上げます。			〒106-0032	東京		
7	見積金額	¥99,000			T E		
8							
9	NO	商品NO	商品名	単価	数量	金額	
10		CH2001	事務用チェア	¥20,000	3	¥60,000	
11		WA3001	ワゴン (3段)	¥10,000	3	¥30,000	
12							
13							
14							
15							

③Set meisaiRng=Range(B10:E15)

明細行の入力範囲全体を meisaiRng に代入する

④cnt = Application.WorksheetFunction.CountA(meisaiRng.Columns(1))

meisaiRng(B10~E15)の1列目のデータの個数を数えて変数 cnt へ代入する

⑤cnt が 0 の場合、メッセージを表示して処理を終了する

⑥Set meisaiRng = meisaiRng.Resize(cnt)

cnt が 0 でない場合、変数 meisaiRng の行数を cnt（ここでは「2」）に設定変更してデータが入力されている範囲だけを取得し、⑦～⑧それぞれのセルの中で空欄がないかチェックする（空欄があればメッセージ文を表示して処理を終了する）

入力データのチェックが終わったら、データを転記する前に、**一覧表に同じ見積NOがあるかどうかを確認**します。

ここでは、見積 No の重複をチェックする部分を作成します。

同じNOが見つかった場合はデータを置き換えるかどうかを確認します。置き換えない場合は重複するセルを選択して終了し、置き換える場合は一覧表で重複するデータを削除します。

【Sample】 [データ転記] プロシージャ ②見積 NO の重複チェック

```
Sub データ転記()
'その① 空欄チェック*****
'その② 見積 No重複チェック*****

    Dim mNo As Long      '見積 Noチェック用
    Dim rng2 As Range    '見積 No重複セル格納用
    Dim ans As Integer   'msgbox 関数の戻り値格納用
    Dim r As Integer     '重複番号の行数取得

    mNo = Range("F2").Value ----①

    With Range("見積 NO").CurrentRegion
        Set rng2 = .Columns(1).Find(what:=mNo, lookat:=xlWhole) ----②
        If Not rng2 Is Nothing Then ----③
            ans = MsgBox("同じ見積番号があります。データを更新しますか?", _
                vbYesNo, "見積番号の重複") ----④
            If ans = vbNo Then ----⑤
                MsgBox "転記をキャンセルします。"
                Application.Goto rng2
                Exit Sub
            Else
                r = Application.WorksheetFunction.CountIf(.Columns(1), mNo) ----⑥
                rng2.Resize(r, 9).Delete xlShiftUp ----⑦
            End If End If
        End With

        MsgBox "重複 NOチェック完了" ----⑧
    'その③ データ転記*****
End Sub
```

- ①セル **F2** の値（見積 NO の数値）を変数 **mNo** に代入する。
- ②名前「見積 NO」のセル（[見積一覧] シートのセル **A3**）を含む表について、**1**列目の中で変数 **mNo** の値を検索し、見つかったセルを変数 **rng2** に代入する。
- ③変数 **rng2** が **Nothing** でない場合（重複データのセルが見つかった場合）、以下の処理を実行する。
- ④更新確認メッセージを表示する。
- ⑤ [いいえ] ボタンがクリックされた場合
→キャンセルメッセージを表示後、変数 **rng2** のセルを選択して終了する。

⑥そうでなかった場合（「はい」ボタンがクリックされた場合）

→見積一覧表の 1 列目の中で、変数 **mNo** と同じ数値の個数を数えて変数 **r** に代入する。

⑦変数 **rng2** のセルを基準に **r** 行 9 列のセル範囲を削除後、下のセルを上へシフトする。

⑧処理の終了を確認するためにメッセージを表示する（完成時は削除する）。

■見積 NO の重複が見つかり、データを更新する場合の処理

① `mNo=Range("F2")`

② `Set rng2 = Range("見積NO").CurrentRegion.Columns(1).Find(what:=mNo, lookat:=xlWhole)`

見積一覧の 1 列目で mNo（4）を探して見つかったセルを rng2 に代入する

`rng2.Resize(r, 9) : セルrng2を基準に r 行、9 列のセル範囲`

⑥ `r = Application.WorksheetFunction.CountIf(.Columns(1), mNo)`

⑦ `rng2.Resize(r, 8).Delete xlShiftUp`

重複データを更新（置き換える）場合は、mNoと同じ値をもつセルを数えて r に代入し（3）、重複データ（`rng2.Resize(r, 9)`まで）を削除する

見積書のデータを見積一覧表に転記していきます。

データの転記先の先頭行を求め、見積 NO、日付、取引先、取引先 NO の転記と明細行の転記を分けています。

【Sample】 [データ転記] プロシージャ ③データ転記部分

Sub データ転記()

「その① 空欄チェック*****

Dim cnt As Long '明細件数代入用 その①、③で使用

Dim meisaiRng As Range '明細入力範囲代入用 その①、③で使用

: (省略)

「その② 見積 NO重複チェック*****

「その③ データ転記*****」

Dim fRow As Long 'データ転記先頭行用

```
Dim copyRng1 As Variant, copyRng2 As Variant
```

```
Dim dataRng1 As Range, dataRng2 As Range
```

'A : 見積 NO、日付、取引先、取引先 NO の転記

With Worksheets ("見積書")

```
copyRng1 = Array(.Range("F2").Value, .Range("F3").Value, _  
                .Range("C5").Value, .Range("A4").Value)
```

```
copyRng2 = meisaiRng.Resize(, 5).Value
```

'明細行 ---- ②

End With

'B : 明細行の転記

With Worksheets ("見積一覧")

```
fRow = .Range("A" & .Rows.Count).End(xlUp).Offset(1).Row ---- ③
```

```
Set dataRng1 = .Range(.Cells(fRow, "A"), .Cells(fRow + cnt - 1, "D")) -----④
```

```
Set dataRng2 = .Range(.Cells(fRow, "E"), .Cells(fRow + cnt - 1, "I")) ----⑤
```

```
dataRng1.Value = copyRng1 ----- ⑥
```

```
dataRng2.Value = copyRng2 ----- ⑦
```

```
.Select
```

```
Range("見積 NO").Sort Key1:=Range("A3"), _ ⑧  
Order1:=xlAscending, Header:=xlYes
```

```
Range("見積 NO").Select
```

End With

End Sub

見積 NO、日付、取引先、取引先 NO を転記します。

'A : 見積 NO、日付、取引先、取引先 NO の転記

With Worksheets ("見積書")

```
copyRng1 = Array(.Range("F2").Value, .Range("F3").Value, _ ①  
                .Range("C5").Value, .Range("A4").Value)
```

```
copyRng2 = meisaiRng.Resize(, 5).Value '明細行 -----②
```

End With

① [見積書] シートのセル**F2** (見積NO)、**F3** (見積日)、**C5** (取引先)、**A4** (取引先NO) の値を、配列変数**copyRng1**に代入する。

②変数**meisaiRng** のセル範囲を列数を**5** 列にサイズ調整したセル範囲の値を、配列変数**copyRng2** に代入する。

(変数**meisaiRng**には商品から数量までの明細行が代入されているため、列数を5に変更して金額まで含めて、変数**CopyRng2**に代入している。)

①変数 copyRng1 に代入する
`copyRng1=Array("5","2025 年 07 月 29 日","C1004","株式会社天空産業")`

	A	B	C	
1	見 積 書			
2			見積 NO	5
3			見積日	2025年07月29日
4	株式会社天空産業		様	
5		C1004	株式会社 SB〇〇商事	
6	下記の通りお見積り申し上げます。			〒106-0032 東京都港区六本木 x-x-x
7	見積金額	¥99,000	TEL: 03-x-x-x-x-x-x	
8				
9	NO	商品NO	商品名	単価 数量 金額
10	1	CH2001	事務用チェア	¥20,000 3 ¥60,000
11	2	WA3001	ワゴン (3段)	¥10,000 3 ¥30,000
12				
13				
14				
15				
16			小計	¥90,000
17			消費税 (10%)	¥9,000
18			合計	¥99,000

②meisaiRng のセル範囲 (商品 NO から数量まで) の列数を 5 にサイズ変更した範囲 (金額列まで) の値を配列変数 copyRng2 に代入する

ブロック B:

明細行を転記します。

'B : 明細行の転記

With Worksheets ("見積一覽")

```
fRow = .Range("A" & .Rows.Count).End(xlUp).Offset(1).Row --- ③
```

```
Set dataRng1 = .Range(.Cells(fRow, "A"), .Cells(fRow + cnt - 1, "D"))④
```

```
Set dataRng2 = .Range(.Cells(fRow, "E"), .Cells(fRow + cnt - 1, "I"))⑤
```



```

dataRng1.Value = copyRng1 -----⑥
dataRng2.Value = copyRng2 -----⑦

.Select
Range("見積 NO").Sort Key1:=Range("A3"), _
    Order1:=xlAscending, Header:=xlYes
Range("見積 NO").Select
End With
End Sub

```

- ③ [見積一覧] シートの **A** 列の最終行から上方向に見ていき、最初に見つかるデータの
 いったセル（見積一覧表の最終行）の **1** 下下のセル（新規入力行）の行番号を変数
fRow に代入する。
- ④ **A** 列の **fRow** 行目から、**D** 列の「**fRow+cnt-1**」行目までのセル範囲を変数
dataRng1 に代入する。
- ⑤ **E** 列の **fRow** 行目から、**I** 列の「**fRow+cnt-1**」行目までのセル範囲を変数
dataRng2 に代入する。
- ⑥ 配列変数 **copyRng1** の値を **dataRng1** のセル範囲に入力する。
- ⑦ 配列変数 **copyRng2** の値を **dataRng2** のセル範囲に入力する。
- ⑧ 見積一覧表の見積 NO を基準に昇順で並べ替える。

	A	B	C	D	E	F	G	H	I
1	見積一覧								
2									
3	見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
4	1	2025/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
5	1	2025/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
6	2	2025/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
7	3	2025/07/28	C1003	株式会社マクスワイ	1001	事務用デスク	¥100,000	1	¥100,000
8	3	2025/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
9	3	2025/07/28	C1003	株式会社マクスワイ	BT4001	パーテーション	¥9,000	1	¥9,000
10	4	2025/07/28	C1002	トゥデイ工業株式会社	cells(fRow,"E")	ア	¥20,000	1	¥20,000
11	4	2025/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
12	4	2025/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
13	5	2025/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
14	5	2025/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

Diagram labels:

- fRow**: points to cell A13 (row 13)
- dataRng1**: points to range D4:D14
- dataRng2**: points to range E4:I14
- cells(fRow+cnt-1,"D")**: points to cell D7
- cells(fRow+cnt-1,"I")**: points to cell I7

- [検索]ボタンで指定した見積 NO のデータを一覧から読み込む

[検索] ボタンをクリックすると、名前「検索NO」のセルに入力された見積書NOのデータを見積一覧の中で検索し、見つかったら、見積書に読み込みます。

名前「検索 NO」

①クリックする

②「検索 NO」の値を検索する

見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
1	2025/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
2	2025/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2025/07/28	C1003	株式会社マクスウィ	DE1001	事務用デスク	¥15,000	5	¥75,000
3	2025/07/28	C1003	株式会社マクスウィ	CH2001	事務用チェア	¥20,000	5	¥100,000
3	2025/07/28	C1003	株式会社マクスウィ	PT4001	パーティション	¥9,000	1	¥9,000
4	2025/07/28	C1002	トッデイ工業株式会社	CH2001	事務用チェア	¥20,000	1	¥20,000
4	2025/07/28	C1002	トッデイ工業株式会社	WA3001	ワゴン (3段)	¥10,000	1	¥10,000
4	2025/07/28	C1002	トッデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000
5	2025/07/29	C1004	株式会社天空産業	CH2001	事務用チェア	¥20,000	3	¥60,000
5	2025/07/29	C1004	株式会社天空産業	WA3001	ワゴン (3段)	¥10,000	3	¥30,000

③見つかったデータを読み込む

NO	商品NO	商品名	単価	数量	金額
1	DE1001	事務用デスク	¥15,000	2	¥30,000
2	CH2001	事務用チェア	¥20,000	2	¥40,000

【データ検索】プロシージャは次ページのとおりです。

【Sample】 [データ検索] プロシージャ

```
Sub データ検索()  
    Dim srcNo As Long      '検索値用  
    Dim srcRng As Range    '売上一覧で検索値を持つセル用  
    Dim srcRow As Long     '売上一覧で検索値を持つセルの行番号用  
    Dim cnt2 As Long       '売上一覧で同じ検索値を持つセルの個数用  
    Dim sData As Variant   '商品データ範囲を配列にして格納用  
    srcNo = Range("検索 NO").Value ---- ①  
    Set srcRng = Range("見積 NO").CurrentRegion.Columns(1).Find( _ ②  
        what:=srcNo, lookat:=xlWhole)  
    If srcRng Is Nothing Then  
        MsgBox "該当する見積 NO のデータがありません" ③  
        Exit Sub  
    End If  
    Call データクリア      'データを読み込む前にデータを削除 ---- ④  
    srcRow = srcRng.Row     '見つかったセルの行番号を調べる ---- ⑤  
    cnt2 = Application.WorksheetFunction.CountIf( _ ⑥  
        Range("見積 NO").CurrentRegion.Columns(1), srcNo)  
    With Worksheets("見積一覧")  
        Range("F2").Value = .Cells(srcRow, "A").Value      '見積 No  
        Range("F3").Value = .Cells(srcRow, "B").Value      '受注日  
        Range("A4").Value = .Cells(srcRow, "D").Value      '取引先名  
        Range("C5").Value = .Cells(srcRow, "C").Value      '取引先 NO  
        sData = .Range(.Cells(srcRow, "E"), .Cells(srcRow + cnt2 - 1, "H")).Value ⑦  
    End With  
    Range(Cells(10, "B"), Cells(10 + cnt 2-1, "E")).Value = sData ---- ⑧  
End Sub
```

①名前「検索 NO」の値を変数 **srcNo** に代入する。

②名前「見積 NO」のセルを含む表（見積一覧表）の **1** 列目の中で変数 **srcNo** の値を検索し、見つかったセルを変数 **srcRng** に代入する。

③見つからなかった場合、メッセージを表示して処理を終了する。

④見つかった場合、[データクリア]プロシージャを呼び出して入力欄をリセットする。

- ⑤変数srcRowに見つかったセルの行番号を代入する。
- ⑥見積一覧表の1列目で変数srcNoと同じ値をもつ数値の個数を調べ、変数cnt2に代入する。
- ⑦見積一覧表のsrcRow 行のA、B、D、C列の値を、それぞれ見積書のセルF2（見積NO）、F3（受注日）、A4（取引先名）、C5（取引先）に入力する。
- ⑧見積一覧表のE列srcRow 行目から、H列「srcRow+cnt2-1」行目までのセル範囲の値を変数sDataに代入する。
- ⑨B列の10 行目からE列の「10+cnt2-1」行目までのセル範囲に変数sDataの値を入力し、明細行のデータを表示する。

<Memo>

「見積一覧」シートのセルA3に名前「見積NO」が設定されています。

⑤srcRow=srcRng.Row

② Set srcRng = Range("見積 NO").CurrentRegion.Columns(1).Find
(what:=srcNo, lookat:=xlWhole)

見積NO	見積日	取引先NO	取引先名	商品NO	商品名	単価	数量	金額
1	2025/07/28	C1001	あかさ商事株式会社	DE1001	事務用デスク	¥15,000	2	¥30,000
1	2025/07/28	C1001	あかさ商事株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
2	2025/07/28	C1005	AAA商会株式会社	CH2001	事務用チェア	¥20,000	2	¥40,000
3	2025/07/28	C1003	株式会社マクスワイ	DE1001	事務用デスク	¥15,000	5	¥75,000
3	2025/07/28	C1003	株式会社マクスワイ	CH2001	事務用チェア	¥20,000	5	¥100,000
3	2025/07/28	C1003	株式会社マクスワイ	PT4001	パーティション	¥9,000	1	¥9,000
4	2025/07/28	C1002	トゥデイ工業株式会社	Cells(srcRow, "E")	事務用チェア	¥20,000	1	¥20,000
4	2025/07/28	C1002	トゥデイ工業株式会社	WA3001	ワゴン Cells(srcRow + cnt2 - 1, "H")			
4	2025/07/28	C1002	トゥデイ工業株式会社	WA3002	ワゴン (4段)	¥14,000	1	¥14,000

⑧ sData = .Range(.Cells(srcRow, "E"), .Cells(srcRow + cnt2 - 1, "H")).Value

⑥ cnt2 = Application.WorksheetFunction.CountIf(Range("見積 NO").CurrentRegion.Columns(1), srcNo)

見 積 書					
			見積NO		1
			見積日		2025年07月28日
あかさ商事株式会社	様				
	C1001		株式会社	SB〇〇商事	
下記の通りお見積り申し上げます。					
見積金額	¥77,000				
NO	商品NO	商品名	単価	数量	金額
1	DE1001	事務用デスク	¥15,000	2	¥30,000
2	CH2001	事務用チェア	¥20,000	2	¥40,000

● [印刷]ボタンで印刷設定をして見積書を印刷する

[印刷] ボタンをクリックしたら、見積書を印刷設定して印刷します。

見積書

見積NO: 5
見積日: 2025年07月29日

株式会社天空産業
C1004
〒106-0032 東京都港区六本木 x-x-x
TEL: 03-x-x-x-x-x-x

見積金額: ¥99,000

NO	商品NO	商品名	単位	数量	金額
1	CH2001	事務用チェア		3	¥60,000
2	WA3001	ワゴン (3段)		3	¥30,000
小計					¥90,000
消費税 (10%)					¥9,000
合計					¥99,000

印刷

①クリックする



見積書

見積NO: 5
見積日: 2025年07月29日

株式会社天空産業
C1004
〒106-0032 東京都港区六本木 x-x-x
TEL: 03-x-x-x-x-x-x

見積金額: ¥99,000

NO	商品NO	商品名	単位	数量	金額
1	CH2001	事務用チェア		3	¥60,000
2	WA3001	ワゴン (3段)		3	¥30,000
小計					¥90,000
消費税 (10%)					¥9,000
合計					¥99,000

〔見積書印刷〕プロシージャは、印刷範囲の設定をしてから印刷を実行します。印刷設定をするには、**PageSetupプロパティ**でページ情報を設定します。

ここでは、**印刷範囲、用紙サイズ、用紙の向き、印刷倍率、ページ中央設定**をして、**プレビューを表示したのち印刷を実行**します。

【Sample】〔見積書印刷〕プロシージャ

```
Sub 見積書印刷()  
    Dim ws As Worksheet  
    Set ws = Worksheets("見積書")  
    With ws.PageSetup  
        .PrintArea = Range("A1:F18").Address --- ①  
        .PaperSize = xlPaperA4 --- ②  
        .Orientation = xlPortrait --- ③  
        .Zoom = False  
        .FitToPagesTall = 1  
        .FitToPagesWide = 1  
        .CenterHorizontally = True --- ⑤  
    End With  
    ws.PrintOut Preview:=True --- ⑥  
    Set ws = Nothing  
End Sub
```

① 〔見積書〕シートについて、印刷範囲をセル **A1** から **F18** に設定する。

② 用紙サイズを **A4** に設定する。

③ 印刷の向きを縦置きに設定する。

④ 用紙の縦、横方向に **1** ページで収まるように設定する。

⑤ 水平方向に用紙の中央に配置する。

⑥ 印刷プレビューを表示する設定で印刷を実行する。